GRAPH DATABASES

Team: Happy?

Ajitesh Jamulkar - 22CS1004
Pallav Agarwal - 22CS30040
Saras Deepak Wagh - 22CS30048
Hemant - 22CS30029

Abstract

This project proposes developing a social media platform that leverages the capabilities of graph databases to model complex relationships and interactions inherent in social networks. The core objective is to demonstrate the efficiency and flexibility of graph databases, specifically Neo4j, in handling interconnected data compared to traditional SQL-based systems.

Work Plan and Methodology

1. Graph Database Implementation:
   The platform will be built using Neo4j as follows
   - Nodes : Users, posts, comments, and activities .
   - Edges : Friends of a user, posts of a user, hashtags of posts.

2. Feature Implementation(Tentative):

- Basic person features
  - Sign Up or Registering a person
  - Edit Person Details
  - Delete Account
  - Login with username and password

- Friends
  - Search Users and send friend requests
  - Accept or Reject Friend Request
  - Able to see friend’s posts and like and comment on it

- Posts
    - User Able to create post
    - Can specify visibility if the post is private, visible to friend or visible to everyone
    - Can search every posts containing a given hashtag
    - Delete Posts

- See Top Influencers Page Rank of a node

3. Custom Dataset Development:
   Stanford SNAP large graph repository consists of **inconsistency in its datasets**. The dataset values are **anonymised and not very realistic** and most of the dataset are just a text file with relation between IDs but no information about nodes. It is not structured enough to use in the project. So to ensure data consistency and relevance, a custom dataset will be generated , simulating user profiles, interactions, and content. Tools like Mockaroo , Faker or custom scripts will be used to create the dataset.

4. Entity-Relationship Structure(Tentative):

    - Nodes: Labeled as User, Post, Comment, Group, Hashtag.
        - Properties include unique IDs, and specific metadata like username, password, Phone, Image, Friend Requests and liked Posts for User

    - Edges: Labelled as FOLLOWS, LIKES, COMMENTED_ON, MEMBER_OF.
        - Edge properties like unique Ids, blocked Users by a user etc.

5. Why Neo4j:
   Unlike SQL, which relies on resource-intensive joins for relationship-heavy queries, Neo4j’s index-free adjacency enables O(1) traversal of connections. For example, fetching a user’s entire network in a social graph requires a single query in Neo4j, while SQL would need multiple nested joins, leading to slower performance as the dataset scales. And Neo4j proves best in terms of functionalities and resources available on the internet to learn and implement the project.

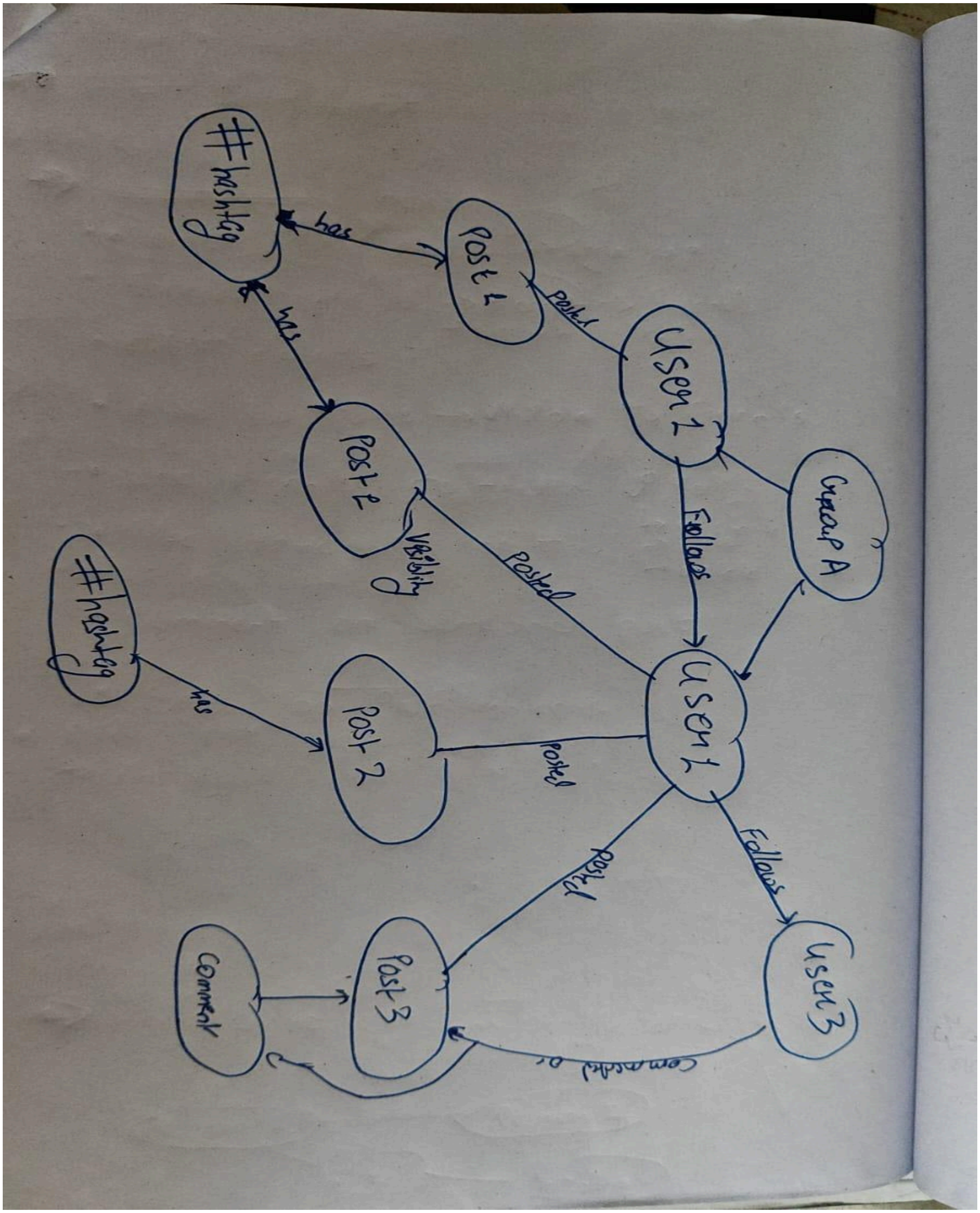
Conclusion:
By implementing a feature-rich social media platform, this project underscores the suitability of graph databases for managing interconnected data. The use of Neo4j, coupled with a purpose-built dataset and performance analysis, will provide actionable insights into the strengths of graph-based architectures in real-world applications.

Weekly work plan

1. Week 1 (17/03/2025 - 24/03/2025):
    - ○ Getting familiar with Neo4j and its cypher language by internet resources
    - ○ Creating small sample databases for the project
2. Week 2 (25/03/2025 - 31/03/2025):
    - ○ Building Frontend: Build the layout and frontend of the project with basic functionalities with some dummy data and making components, UI, animations by tailwind and framer motion in Next.js.
    - ○ Building Backend : Build basic the backend functionalities like functions for connecting to Neo4j database, APIs for CRUD - fetch query results, create nodes and edges, update and delete
3. Week 3 (01/04/2025 - 07/04/2025):
    - ○ Building Frontend: Finish the frontend till a fully functional prototype with all the pages and button functionalities.
    - ○ Integration: Integrating frontend with backend functions and calling Apis to fill the dummy data with correct data and making Button functionalities
4. Week 4 (08/04/2025 - 14/04/2025):
    - ○ Testing functionalities and Bug Fixes and populating database with actual large graph dataset

- ○ Try to build advanced features like support for stories and chats between users, notifications of friend requests etc.

5. Visual representation of database -


#hashtag
has
Post 1
posted
User 1
has
Post 2
Group A
Follows
Visibility
Posted
#hashtag
has
Post 2
posted
User 2
Posted
Follows
Comment
Post 3
User 3
commented on